Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Blaž Koncilja

Napovedovanje porabe pomnilniških kapacitet pri rezervnem kopiranju

diplomsko delo
univerzitetni študij

Mentor: prof. dr. Marko Robnik-Šikonja

Ljubljana 2016

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Za ponudnike in upravnike računske in pomnilniške infrastrukture je pomembno, da znajo natančno predvideti porabo posameznih virov. Rezervno kopiranje je velik porabnik pomnilniških kapacitet, zato morajo ponudniki tovrstnih storitev pravočasno zagotoviti dovolj virov in se pripraviti na nastop povečanih zahtev, morebitnih ozkih grl in izjemnih dogodkov. Analizirajte podatke o porabi pomnilniških kapacitet ponudnika storitev rezervnega kopiranja in poskušajte napovedati porabo za določen čas naprej. Uporabite napovedne metode in sledite standardni metodologiji podatkovnega rudarjenja.

Izjava o avtorstvu diplomskega dela

Spodaj podpisani Blaž Koncilja, z vpisno številko **63080014**, sem avtor diplomskega dela z naslovom:

Napovedovanje porabe pomnilniških kapacitet pri rezervnem kopiranju

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom prof. dr. Marka Robnika-Šikonje,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 30. avgust 2016 Podpis avtorja:

Za nasvete in pomoč se zahvaljujem mentorju prof. dr. Marku Robniku-Šikonji. Zahvaljujem se vsem sodelavcem, ki so mi pomagali z idejami in lastnimi izkušnjami iz tematike naloge. Zahvaljujem se tudi družini, ki me je spodbujala med pisanjem diplomske naloge in celotno študijsko pot.

Kazalo

Povzetek

Naslov: Napovedovanje porabe pomnilniških kapacitet pri rezervnem kopiranju

Potreba po rezervnem kopiranju oziroma arhiviranju podatkov v svetu narašča. Podjetja potrebujejo za normalno delovanje hraniti vedno več informacij. Hranjenje teh podatkov lahko predstavlja velik strošek, zato hočemo pomnilniško kapaciteto držati na ravni, ki zadovolji naše potrebe in hkrati ni predimenzionirana.

S pomočjo podatkovnega rudarjenja želimo napovedati trende porabe pomnilniških kapacitet. Najprej smo pridobili podatke iz dveh različnih okolij za arhiviranje in jih shranili v podatkovno bazo. To nam je omogočilo hitro združevanje in upravljanje s podatki. Podatke smo analizirali z metodami linearne regresije, linearne regresije po kosih in k-najbližjih sosedov. Za najbolj zanesljivo metodo za napovedovanje trendov se je izkazala linearna regresija po kosih.

Čeprav so rezultati dovolj dobri za uvedbo metode v produkcijo, moramo biti previdni, saj sta se analizirani okolji izkazali za zelo različni, kar neposredno vpliva na zanesljivost napovedi.

Ključne besede: podatkovno rudarjenje, postopek CRISP-DM, linearna regresija po kosih, priprava podatkov

Abstract

Title: Forecasting backup storage consumption

Storage needs for archiving data are increasing. Companies need to store more and more data to function normally. Storing this data can be costly, that is why we want to provide sufficient storage capacity to meet the demands and not exceed them which brings additional costs.

With the help of data mining we are trying to forecast trends in storage consumption. We acquired data from two environments for archiving and saved them to a database. We analysed data consumption trends with linear regression, piecewise linear regression and k-nearest neighbours. Piecewise linear regression proved to be the most accurate and reliable.

Even though results are good enough to be implemented into production, we should be cautious as the two environments have different characteristics and this influences the forecasting.

Keywords: data mining, procedure CRISP-DM, piecewise linear regression, data preparation

1. Uvod

Napovedovanje trendov v podatkovnem arhiviranju je, še posebej za velika okolja, pomembno z več vidikov. Prvi je predvsem stroškoven, opremo želimo čim bolj optimalno izkoristiti in hkrati nočemo predimenzionirati arhivskega okolja. Drugi pa je, da s poznavanjem trendov lahko predvidimo probleme ter jih tudi preprečimo .

Na trgu imamo na voljo kar nekaj rešitev za arhiviranje podatkov velikih in priznanih proizvajalcev. Segmentu se napoveduje konstantna rast v prihodnosti. Število programov, ki napovedujejo trende v okolju za arhiviranje, pa ni veliko To zmožnost v svoje produkte proizvajalci uvajajo šele zadnjih nekaj let.

Znanstvenih člankov, ki se ukvarjajo s področjem podatkovnega rudarjenja ne manjka [9]. Raziskave potekajo več desetletij in akademiki odkrivajo vedno nova področja, kjer lahko uspešno odkrivajo nova dognanja iz obstoječih podatkov. Eno takih področji je biomedicina. Na področju podatkovnega arhiviranja ne najdemo veliko člankov, večina raziskuje podatkovne centre, predvsem z vidika omrežne povezljivosti in porabe energije. Znanstvenih del, ki bi se ukvarjale z napovedjo trendov na podlagi statistike in značilnostih arhiviranih podatkov, je glede na to kako živahno je področje podatkovnega rudarjenja, relativno malo [2]. Objavljajo jih predvsem strokovnjaki, zaposleni v vodilnih podjetjih na tem področju. To kaže, da je raziskav verjetno več, ampak niso dostopne javnosti, saj znanje podjetja zadržujejo zase, ker jim prinaša konkurenčno prednost.

Cilj diplomske naloge je, da v okolju za arhiviranje iz podatkov, ki so nam

na voljo, pridemo do novih ugotovitev in napovemo trende.Omejili smo se na področje kapacitete, napovedujemo celotno količino shranjenih podatkov za prihodnost po dnevih.

Analizo smo naredili po postopku CRISP-DM (Cross Industry Standard Process for Data Mining) [3]. Na izbiro je vplivalo, da ta proces, bolj kot preostali, zajema tudi poslovni vidik. Poleg tehničnih korakov podatkovnega rudarjenja, kot so priprava podatkov in izgradnja modelov, pokrijemo tudi vprašanje, katero poslovno potrebo hočemo pokriti s podatkovnim rudarjenjem. S tem postopek podatkovnega rudarjenja dobi jasen cilj.

Za napovedovanje trendov uporabljamo algoritma linearne regresije in k-najbližjih sosedov. Velik del dela zajema zbiranje in priprava podatkov. Informacije zbiramo iz podatkov o delovanju programa, ki opravlja arhiviranje. Podatki nam niso bili prosto na voljo, ampak smo jih morali pridobiti iz produkcijskega okolja. Napovedane algoritme smo implementirali v programskem jeziku java, uporabili smo knjižnico WEKA [6].

V prvem poglavju opišemo problem napovedovanja trendov v okolju za arhiviranje, kot ga razlaga trg. Pregledali smo že obstoječe programske rešitve ter podrobneje analizirali strokovni članek, ki se ukvarja s področjem napovedovanja kapacitete. V drugem poglavju opišemo postopek podatkovne analize CRISP-DM in ostale uporabljene tehnologije. V tretjem poglavju izvedemo akcije, ki smo jih opisali v drugem poglavju. Pripravimo podatke in izgradimo modele. Modele najprej ocenimo v tehničnem smislu s pomočjo uveljavljenih metod, potem pa jih ocenimo še glede na rezultate ki jih želimo doseči v realnem okolju. Naredimo načrt kako bi ugotovljene rešitve vpeljali v produkcijo. V zaključku opišemo kaj smo v diplomskem delu naredili in rezultate našega dela kritično ocenimo. Končamo z idejami za nadgradnjo opravljene raziskave.

2. Napovedovanje trendov v okolju za podatkovno arhiviranje

Trg se zaveda, da je napovedovanje trendov v okolju za arhiviranje pomembno področje tako s stroškovnih razlogov, kot tudi za zagotavljanje optimalnega poslovanja. Veliko IT (Informacijska tehnologija) podjetij se ukvarja z nudenjem storitev na področju postavitve in upravljanja podatkovnih centrov in okolja za arhiviranje. Eno od takih podjetji, Signature Technology Group pravi, da mora rešitev za upravljanje okolij pokrivati tri področja:

1. Ocenjevanje. Razumeti moramo naše poslovne potrebe. Upravljanje s kapaciteto okolja je proces, ki mora zreti v prihodnost. Za to moramo dobro poznati poslovne potrebe , ki jih naše okolje pokriva.

2. Nadziranje. Nadzirati moramo ali okolje zadostuje operacijam, ne smemo dovoliti, da okolje preprečuje dosego poslovnih ciljev.

3. Optimizacija izrabe. Sredstva okolja za hranjenje podatkov lahko zajema dobršen del porabe IT virov v podjetju, zato je naš cilj da so izkoriščeni optimalno [16].

2.1 Obstoječe rešitve

Glede na pomemnost področja imamo na trgu že nekaj rešitev, ki obljubljajo napovedovanje vrednosti iz obstoječih podatkov. Eden takih je OpStor, podjetja ManageEngine [12], ali pa Storage Resource Monitor, podjetja SolarW-

inds [17]. Orodja na osnovi obstoječih podatkov o zasedenosti v določenih časovnih obdobjih izračunajo napoved za prihodnost. Noben program ne razkrije, katere napovedne modele so uporabili, kar je, glede na to da gre za komercialne produkte, razumljivo. Noben od produktov tudi ne pove, kakšno uspešnost dosega pri napovedih, in ali odkrije svojo neuspešnost pri napovedih in jo sporoči uporabniku.

2.2 Obstoječi pristopi

Znanstvenih del, v katerih bi pokrivala področje napovedovanja trendov v okolju za arhiviranje, je malo. Glede na to, da je člankov glede upravljanja kapacitete v strežniških okoljih kar nekaj, je to nenavadno. Dober opis daje članek [2], ki ga obširneje povzemamo v nadaljevanju tega razdelka. Motivacija za izgradnjo napovednega modela je ugotovitev, da IT organizacije prepogosto delujejo reakcijsko, torej ukrepajo šele, ko njihov sistem doseže polno kapaciteto. To pomeni, da so že nastopile zmogljivostne težave, ali celo izguba podatkov. Zaradi tega nastane potreba po orodju, ki napoveduje rast in opozorja, preden se okolje zapolni.

Za zbiranje podatkov imajo avtorji zelo dobro rešitev. Njihov produkt za arhiviranje ponuja storitev 'avtomatska podpora'. Vsak dan se iz okolij strank pošilja diagnostične informacije o sistemu na centralni strežnik podjetja. To jim, poleg tega, da nudijo boljšo podporo strankam, omogoča, da lahko opravijo statistične analize in postopke podatkovnega rudarjenja na veliki količini realnih podatkov. Podatke so tudi dodatno prečistili. Za nas je zanimiva informacija, da so odstranili testne podatke, ker se zbirajo tudi podatki znotraj podjetja, ko testirajo in razvijajo produkte. Mi imamo podoben primer, ki se med razvojem uporablja skoraj vedno, uporabljajo pa ga tudi stranke. Arhiviranje lahko izvedeš z namišljeno napravo ter s tem preizkusiš vse nastavitve in omrežje.

Model, ki so ga izbrali v članku [2], je linearna regresija. Algoritem se je slabo izkazal v primeru večje spremembe trendov, zato so izvedli prilagajanje

izvornih podatkov tako, da so izbrali manjši nabor novejših podatkov. Vzeli so zadnjih deset primerov podatkov in zgradili model, potem pa so vzeli še enajsti primer in zgradili model, ter tako naprej. Za vsak model so izračunali kakovost napovedi v obliki R^2. Za napovedovanje so izbrali model linearne regresije z najboljšim rezultatom R^2. Poleg R^2 so pri izbiri končnega modela upoštevali še druge pogoje, od katerih sta najpomembnejša dva:

1. da je na voljo dovolj podatkov, vsaj petnajst dni.
2. zadnja podatkovna točka je najpomembnejša. Če model daje dobre rezultate, najnovejši primer pa oceni slabo, pomeni, da se je zgodila večja sprememba v sistemu in bodo napovedi lahko nekaj časa slabe. Na primer, sistem se je približevali polni kapaciteti in IT inženir zbriše mnogo starih podatkov, takrat nastane velika sprememba v kapaciteti in je potrebno počakati, da se okolje ponovno normalizira, preden lahko spet pričakujemo dobre napovedi.

Po opisanem postopku so analizirali svoje preostale stranke in prišli do naslednjih rezultatov:

1. povprečje R^2 za vse sisteme je 0.93,
2. 60 % sistemov je imelo $R^2 \geq 0.90$,
3. 78 % sistemov je imelo $R^2 \geq 0.80$

Iz tega so lahko sklepali, da se večina sistemov obnaša linearno. Njihov končni cilj je bil napovedati, kdaj bo sistem dosegel polno zasedenost. Pri tem so opozorili na težave, na katere lahko naletijo.

- Lažno pozitivni rezultati.
- Težave z akcijami, ki jih linearni model ne zazna, kot so spremembe strojne opreme, zamenjava celotnega sistema, dodajanje novih diskov ali spremembe v programski opremi, kot je sprememba, koliko časa hranimo podatke.

- Nimajo napovedi, kdaj bo sistem dosegel polno kapaciteto, kadar linearni model ne da dovolj zanesljive napovedi glede na prej opisane pogoje.

V delu [2] so prišli do zaključka, da bodo samodejno zgrajeni modeli za napovedovanje trendov v IT sistemih postali vedno bolj zaželeni in potrebni z njihovo rastjo v velikosti in kompleksnosti. Na koncu postavijo tri vprašanja za nadaljnje raziskave:

1. Ali obstajajo še druga področja, kjer bi lahko uporabili ta model za podatkovno rudarjenje v obstoječih podatkih, kot so izravnava obremenitve ali prepustnost mreže?

2. Zakaj linearni model na nekaterih primerih da slabe rezultate? Ali lahko izboljšamo obstoječi model ali je kak model boljši?

3. Ali lahko s statistično analizo najdemo korelacije med sistemskimi nastavitvami ali časovno vrsto in drugimi podatki?

2.3 Obstoječi postopki za podatkovno rudarjenje

V uporabi so trije standardizirani procesi za podatkovno rudarjenje: KDD, SEMMA in CRISP-DM [1]. Največ se uporablja CRISP-DM, potem SEMMA, ki je v upadanju, in KDD. Po anketi iz leta 2014, ki so jo opravili pri KDnuggets [14], vidimo, da CRISP-DM uporablja 43 % anketirancev, SEMMA 8,5 % ter KDD proces 7,5 %; ostali uporabljajo svoje metodologije. KDD pristop oziroma Knowledge Discovery in Databases je najstarejši, osnovali so ga že 1989. Pristop začrta splošen proces iskanja novih znanj iz podatkov in poudarja aplikativno vrednost metod podatkovnega rudarjenja. SEMMA je kratica za Sample, Explore, Modify, Model, Assess. Nanaša se na korake, ki jih postopek zajema. Postopek je razvil SAS Institute. Čeprav je postopek

samostojen, se povezuje s programom SAS Enterprise Miner. CRISP-DM oziroma CRoss-Industry Standard Process for Data Mining je leta 1995 definiral konzorcij podjetij Daimler-Benz, SPSS (sedaj del IBM) in NCR Corporation. Postopek je dobro dokumentiran in ima ustrezno organizirane, strukturirane in definirane korake. Vsak od procesov ima več korakov, ki so si enakovredni glede na to, kar definirajo [1]. Kateri koraki so si enakovredni vidimo v tabeli 2.1.

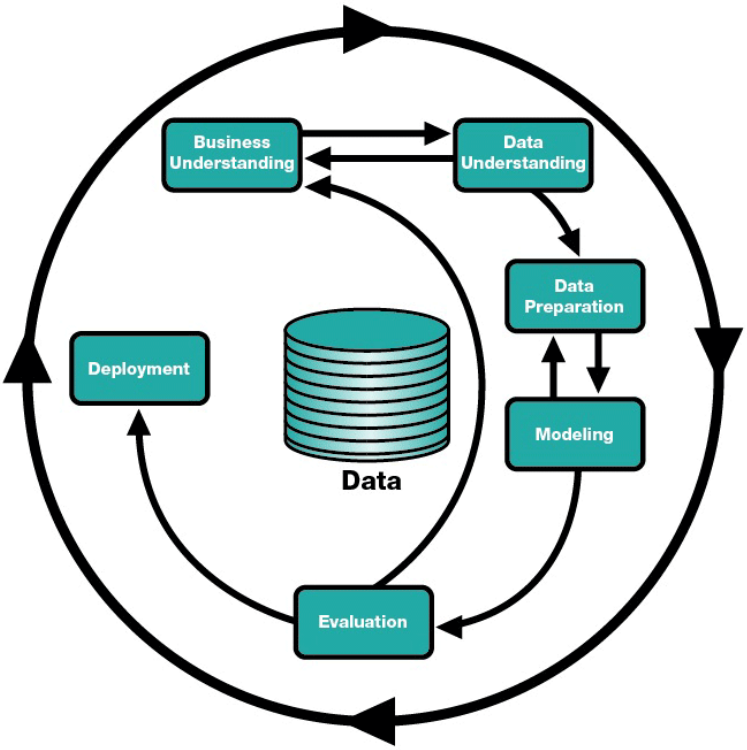
KDD	SEMMA	CRISP-DM
pred KDD	—	razumevanje poslovnega vidika
izbira	vzorčenje	razumevanje podatkov
pred procesiranje	raziskovanje	
transformacija	prilagajanje	priprava podatkov
podatkovno rudarjenje	modeliranje	modeliranje
interpretacija/ocenjevanje	ocenjevanje	vrednotenje
po KDD	—	uvajanje v produkcijo

Tabela 2.1: Prekrivanje korakov med KDD, SEMMA in CRISP-DM.

Uporabili bomo postopek, ki ga definira CRISP-DM, ker je dobro dokumentiran in natančno opisuje korake. Definira korake, ki se ukvarjajo s poslovnim vidikom podatkovnega rudarjenja in je najbolj priljubljen postopek med uporabniki.

2.4 Opis postopka podatkovnega rudarjenja po metodi CRISP-DM

Postopek podatkovnega rudarjenja, ki ga definira CRISP-DM referenčni model, je razdeljen na šest delov. Vrstni red faz ni tog in med posameznimi fazami se lahko premikamo glede na rezultate drugih faz [3]. V nadaljevanju opišemo posamezne korake pikazene na sliki 2.1.



Slika 2.1: Faze referenčnega modela CRISP-DM [10].

2.4.1 Razumevanje poslovnega vidika

2.4.1.1 Določitev poslovnih ciljev

Določimo cilje in jih opišemo s poslovnega stališča. Ugotoviti moramo vse poslovne vidike, ki lahko vplivajo na postopek podatkovnega rudarjenja. Določimo tudi pogoje, ki določajo, kdaj je podatkovno rudarjenje uspešno.

2.4.1.2 Ocena virov

Pripravimo seznam vseh virov, ki jih imamo na voljo za izvedbo. Viri so strojna in programska oprema, podatki, ki so nam na voljo in tudi človeški viri, ki jih lahko izkoristimo. V tem koraku določimo tudi vse omejitve, ki nam jih določajo viri od časovnih do zakonskih. Ocenimo tudi cenovno

vrednost virov in cenovno korist, če so poslovni cilji doseženi.

2.4.1.3 Določitev ciljev podatkovnega rudarjenja

Določimo tudi cilje podatkovnega rudarjenja, pred tem smo določili poslovne cilje. V tej fazi določimo cilje in kriterije za uspeh s tehničnega stališča.

2.4.1.4 Projektni plan

Na koncu ustvarimo projektni načrt. Naštejemo vse korake, ki jih bomo izvedli v postopku podatkovnega rudarjenja, ter ocenimo, koliko sredstev in časa nam bodo vzeli. Za vsako fazo projektni plan vsebuje, katere postopke in strategije nameravamo uporabiti.

2.4.2 Razumevanje podatkov

2.4.2.1 Zbiranje začetnih podatkov

Pridobimo podatke oziroma dostop do podatkov. Podatke naložimo v orodja, v katerih bomo z njimi upravljali. Opišemo tudi vse probleme, na katere smo naleteli med zbiranjem podatkov, in načine, kako bi jih rešili.

2.4.2.2 Opis podatkov

Površinsko opišemo zbrane podatke. Opišemo njihovo obliko, količino. Vprašamo se, ali podatki zadovoljijo naše potrebe.

2.4.2.3 Raziskava podatkov

Tu pričnemo prve korake podatkovnega rudarjenja, ko naše podatke pregledamo in po potrebi grafično prikažemo. Pregledamo in prikažemo porazdelitve atributov, poiščemo relacije med atributi in naredimo osnovne statistične analize, kot je na primer povprečenje. Raziskava podatkov nadgradi osnovni opis podatkov in hrani naslednje korake, predvsem pripravo podatkov.

2.4.2.4 Preverba kvalitete podatkov

Preverimo, ali so naši podatki kompletni, torej ali imamo manjkajoče dele in če pokrivajo vse primere, ki jih analiziramo. Pregledamo podatke za napake in naštejemo rešitve za le-te.

2.4.2.5 PostgreSQL

Zbrane podatke bomo hranili v podatkovni bazi. PostgreSQL, ali skrajšano postgres, je objektno relacijska baza. Prva verzija baze je bila izdana pred dvajsetimi leti. Postgres se razvija odprtokodno. Primerna je za poslovne rešitve, saj omogoča shranjevanje velikih količin podatkov. Tabele lahko dosežejo do 32 TB velikosti in sama baza nima omejitve glede velikosti. Postgres implementira večino standarda ISO/IEC 9075:2011, kar je uporabno, saj je baze, ki se držijo standarda, lažje preselili v drug bazni sistem. Postgres se tudi hvali z eno boljših dokumentacij, kar olajša razvijanje baze. [15]

2.4.3 Priprava podatkov

2.4.3.1 Izbira podatkov

Izmed vseh zbranih podatkov v tem koraku izberemo tiste, ki jih bomo uporabili v analizi. Izbiramo glede na svoje cilje in glede na tehnične omejitve, kot je količina. Ker imamo lahko omejen čas in računsko moč za našo analizo ter glede na izbrane postopke podatkovnega rudarjenja, ne moremo uporabiti vseh podatkovnih tipov.

2.4.3.2 Čiščenje podatkov

V tem koraku poskušamo dvigniti kakovost podatkov glede na izbrane metode podatkovnega rudarjenja. Lahko izberemo manjše sete podatkov, ki so glede na le-te bolj primerni za analizo. Tu se sklicujemo na preveritev kakovosti podatkov iz faze razumevanje podatkov, kjer smo našteli rešitve za napake,

in jih izvedemo. Upoštevati moramo tudi, ali kateri od postopkov v tej fazi vpliva na točnost naših rezultatov.

2.4.3.3 Priprava atributov

Ustvarimo nove atribute oziroma iz obstoječih pridobimo izpeljane. Izpeljane atribute pridobimo tako, da jih izračunamo iz obstoječih atributov. Nove atribute pridobimo tako, da glede na razumevanje obstoječih atributov ustvarimo nove, ki jih v osnovnih podatkih ni.

2.4.3.4 Integracija podatkov

V tem koraku združujemo podatkovne množice. Povežemo več tabel, ki imajo različne informacije o skupnem objektu. Ta korak pokriva izračun agregatov iz obstoječih podatkov, kot sta seštevanje in izračun povprečja.

2.4.3.5 Formatiranje podatkov

Podatke sintaktično spremenimo tako, da ne spreminjamo njihovega pomena. Nekatera orodja pričakujejo atribute v določenem vrstnem redu, ali pa da moramo za vsak podatek imeti unikaten atribut. Orodja in posamezni algoritmi delujejo bolje, če podatke uredimo na določen način. Uporabljena orodja in programske knjižnice imajo specifične omejitve, kot so razmejitve podatkov z določenim ločilom, ali da podatki ne smejo imeti praznega prostora.

2.4.4 Modeliranje

2.4.4.1 Izbira modelirne tehnike

V tem koraku izberemo in opišemo konkretne metode modeliranja, kot so npr. odločitvena drevesa. Tukaj opišemo, če posamezna metoda vključuje predpostavke glede podatkov, ki jih obdeluje.

2.4.4.2 Izdelava načrta za testiranje

Preden zgradimo model, moramo določiti mehanizem, s katerim bomo preizkusili njegovo uspešnost. Če je potrebno, razbijemo množico podatkov na učno in testno množico. Na prvi zgradimo model, na drugi preverimo njegovo napovedno točnost. Definiramo tudi matematične formule za preverjanje točnosti.

2.4.4.3 Izgradnja modela

Izvedemo postopke podatkovnega rudarjenja na zbranih podatkih. Opišemo vse parametre, ki jih lahko vsebuje algoritem, in njihove vrednosti. Opišemo tudi model, ki smo ga zgradili.

2.4.4.4 Ocena modela

Model ocenimo po načrtu za testiranje in glede na cilje podatkovnega rudarjenja, ki smo jih postavili v prvi fazi pri določitvi ciljev podatkovnega rudarjenja. Naredimo revizijo parametrov modela in se vračamo nazaj na izgradnjo, dokler ne dosežemo po naši oceni najboljšega možnega modela.

2.4.4.5 WEKA

Za izgradnjo in oceno napovednih modelov uporabljamo orodje WEKA. WEKA je okrajšava za Waikato Environment for Knowledge Analysis. Gre za zbirko algoritmov strojnega učenja in metod za predprocesiranje podatkov. Omogoča podporo za podatkovno rudarjenje, pripravo podatkov in statistično ocenjevanje učnih shem. Razvija se na Univerzi iz Waikata na Novi Zelandiji. Napisana je v javi in lahko teče na vseh razširjenih operacijskih sistemih, kot so Windows, Linux in Os X. Obstaja več možnosti uporabe. Posamezne algoritme lahko kličemo preko ukazne vrstice, tako da v ukazno vrstico, vpišemo ime datoteke .class, ki vsebuje algoritem. Določimo parametre ukaza, -h prikaže pomoč in izpiše vse možnosti ukaza. WEKA tudi vsebuje grafični uporabniški vmesnik. Vmesnik lahko razdelimo na tri dele:

1. Raziskovalec ('Explorer') vodi, izklaplja in vklaplja možnosti glede na izbrano. Za nekatere algoritme ima že pred nastavljene vrednosti.

2. Tok znanja oziroma 'Knowledge Flow' predstavlja učenje v toku podatkov ('data stream'). Ko določimo podatkovni nabor raziskovalec naloži vse podatke v pomnilnik. Tok znanja omogoča, da izvajamo učenje po korakih na manjših količinah podatkov. Učenje na toku podatkov mora omogočati tudi izbrani algoritem.

3. Tretji grafični vmesnik je eksperimentator 'experimenter'. Z njim lahko avtomatiziramo proces preizkušanja različnih algoritmov. Omogoča nam izvajanje na več računalnikih s pomočjo 'Java remote method invocation'.

Algoritme, implementirane v zbirki WEKA, lahko integriramo v naše aplikacije, spisane v javi. Z vmesnikom programa lahko izvajamo vse korake podatkovnega rudarjenja, ki nam ga omogoča grafični vmesnik oziroma ukazna vrstica. Implementirati moramo štiri glavne komponente:

1. učne primere, v katerih preberemo podatke,

2. filtre, kjer prečistimo podatke,

3. klasifikator, v katerem na naših podatkih izvedemo želen algoritem,

4. ter ocenjevalno komponento, kjer na različne načine ocenimo delovanje algoritma [6].

2.4.5 Vrednotenje

Napovedne modele smo ocenili z različnimi funkcijami za ocenjevanje napovedi. Rezultate komentiramo in poskušamo razložiti. Ocenimo tudi, ali bi lahko katerega od korakov izpeljali drugače in dosegli boljše rezultate ter kako bi delo nadaljevali.

Do sedaj smo vrednotili modele z izračuni mer napake, opisanimi v načrtu za testiranje.V tem koraku pa ocenimo, v kolikšni meri smo pokrili poslovne

cilje podatkovnega rudarjenja in ugotavljamo, ali obstaja poslovni razlog, zaradi katerega bi bil model neuspešen. Ocenimo tudi vse ostale rezultate v postopku podatkovnega rudarjenja, tudi take, ki mogoče niso direktno vezani na osnovne poslovne cilje, a nam pomagajo pri prihodnjih odločitvah.

Glede na ugotovitve med vrednotenjem se odločimo, kako nadaljevati. Odločimo se, ali uvedemo projekt v produkcijo ali se vrnemo nazaj in ponovimo nekatere korake, ali pa s projektom zaključimo.

2.4.6 Uvajanje v produkcijo

Tu definiramo načrt, kako naše ugotovitve uvesti v produkcijo. Povzamemo korake, ki so potrebni, in kako jih izvesti.

Če rezultati podatkovnega rudarjenja vplivajo na vsakdanje procese poslovanja, jih moramo v produkciji primerno nadzorovati in se s tem izogniti daljšim časovnim obdobjem, v katerih bi lahko napačni rezultati vplivali na poslovanje.

Naredimo pregled celotnega postopka napovedovanja porabe pomnilniških kapacitet. Ocenimo kaj smo delali dobro in kaj slabo. Z ugotovljenim pospešimo in olajšamo delo na prihodnjih projektih podatkovnega rudarjenja.

3. Analiza problema

V tem poglavju analiziramo podatke in jih pripravimo za nadaljnjo uporabo. Na njih smo zgradili modele in izračunali njihovo točnost. Modele smo ocenili ali jih lahko uporabimo za napovedovanje porabe pomnilniških kapacitet v testnih okoljih in ali bi jih lahko uvedli tudi v produkcijo.

3.1 Razumevanje poslovnega vidika

Podjetja shranjujejo velike količine podatkov, ki ponavadi naraščajo. Arhiviranje podatkov podjetju pomeni strošek, ki ga želi minimizirati. Eden od načinov, da strošek zmanjša, je, najmanjša možna količina presežnega prostora. V okolju za arhiviranje moramo vedno imeti presežek, ker moramo pokriti trenutne potrebe in zato, ker naprave delujejo počasneje, ko so zapolnjene do konca. Presežek je torej nujen, nočemo pa, da je prevelik ali premajhen.

Uporabniki želijo velikost svojega okolja ohranjati na optimalni ravni glede na njihove potrebe. To lahko dosežejo tako, da imajo zaposlenega ali najetega strokovnjaka IT, ki nadzoruje njihovo okolje. Naš cilj je te stroške zmanjšati s tem, da zagotovimo avtomatizirano rešitev, ki bo potrebovala manj človeške interakcije in bo na ta način na dolgi rok cenejša za uporabnika.

Da izpolnimo poslovne cilje, moramo ustvariti rešitev, ki bo dovolj zanesljivo napovedala trende v arhivskem okolju, s katerimi bomo zmanjšali potrebo inženirjevemu času za nadzorovanje rasti le-tega. Ali smo uspešni, bomo preverili interno v podjetju z IT strokovnjakom, ki skrbi za shranjeva-

nje in arhiviranje podatkov.

Za podatkovno analizo in pripravo rešitve so nam na voljo podatki o arhiviranju v našem podjetju in podatki iz produkcije. Podatki iz notranjega okolja segajo od aprila 2015 do julija 2016. Pridobljeni produkcijski podatki za analizo segajo od novembra 2015 do marca 2016. Čeprav podatki iz notranjega okolja obsegajo daljše časovno obdobje, je podatkov iz produkcijskega okolja približno 50 krat več.

Pri analizi je sodelovalo več strokovnjakov. Poleg avtorja diplomskega dela, ki je zbral in pripravil podatke, preizkusil algoritme za analizo podatkov in ocenil njihovo učinkovitost ter mentorja, ki je ekspert na področju podatkovnega rudarjenja, sodelujemo še s strokovnjakom za prodajo, ki pozna potrebe strank na področju podatkovnega arhiviranja, arhitektom več programskih rešitev s področja podatkovnega arhiviranja in več oseb, ki se ukvarjajo s podatkovnim arhiviranjem: od razvijalcev do svetovalca za izgradnjo in uvajanje rešitev.

Tehnični cilj podatkovnega rudarjenja je napovedati količino shranjenih podatkov za prihodnost. Pripraviti moramo tabelo vrednosti, iz katerih lahko naredimo graf, ki bo na razumljiv način predstavil rast količine stranki. Da bo postopek podatkovnega rudarjenja lahko uspešen, moramo napovedati prihodnjo količino z največ desetodstotno napako. Algoritem mora zaznati, če ni zmožen napovedati prihodnjih dogodkov s tako zanesljivostjo, in to sporočiti. Algoritem mora biti sposoben izračunati rezultate v eni noči, torej mora dokončati procesiranje v največ osmih urah.

3.2 Podatki o arhiviranju

3.2.1 Zbiranje podatkov

Pri zbiranju podatkov iz okolja za arhiviranje nam pomaga program, ki mu bomo dodali analitične zmožnosti. Program zbere podatke iz okolja, kot so število shranjenih podatkov, njihova velikost, čas arhiviranja, kakšna strojna

oprema je opravila arhiviranje, na kakšen medij so se podatki shranili in še nekaj drugih bolj podrobnih informacij.

Pridobljene podatke smo shranili v postgres bazo. Vsebujejo informacije o podatkih, ki se arhivirajo, kot so čas, tip in velikost.

Da bomo zmožni narediti analizo na podatkih, moramo imeti vsaj tri mesece podatkov, saj lahko le tako preverimo svojo uspešnost.

Analizirali bomo na dveh okoljih, produkcijskem in internem v našem podjetju.

3.2.2 Opis podatkov

Produkcijsko okolje.

Podatki med 1. decembrom 2015 in 10. marcem 2016, arhiviranje podatkov, pognano v 538314 sejah, v teh sejah je bilo shranjenih 5662197 objektov.

Okolje v domačem podjetju.

Podatki med 1. januarjem 2016 in 1. julijem 2016, arhiviranje podatkov pognano v 1613 sejah, v teh sejah je bilo shranjeno 4925 objektov.

Tabela sej vsebuje informacije o:

1. Začetek; čas, ko se je seja začela.
2. Konec; čas, ko se je seja končala, upoštevani so tudi čas, ko je seja čakala na vire, prosto napravo za sprejem podatkov.
3. Uporabnik; kdo je sejo definiral, progam ima lahko več uporabnikov, ki so nastavljeni ročno ali z uporabo LDAP.
4. Ime računalnika na katerem teče program za arhiviranje.
5. Statusu seje; seja lahko še teče, je končana, čaka na prosto napravo, lahko je spodletela.
6. Tipu arhiviranja; ali se shranijo vsi podatki ali samo razlike od prejšnjega arhiviranja.

Tabela objektov vsebuje informacije o:

1. Začetek; kdaj se je določen objekt začel arhivirati.
2. Konec; kdaj se je arhiviranje objekta končalo.
3. Velikost objekta v KB.
4. Število datotek; objekt je lahko na primer celoten C disk, torej en objekt zajame večje število datotek, tu izvemo koliko;
5. Število opozoril in napak je število težav med arhiviranjem, na primer, datoteka je bila odprta in program ni mogel dostopati do nje. Da bi ugotovili, kaj se je v resnici zgodilo, bi morali pogledati v poročilo, ki se ustvari na koncu seje.
6. Naprava, ki je opravila arhiviranje, ime računalnika, ki naredi zapis na medij. Na primer zapisovalec na trakove.
7. Tip objekta; lahko so datotečni sistemi Windows ali Unix/Linux ter razne podatkovne baze oziroma strežniku, kot so MSSQL, Oracle, IBM DB2, MS SharePoint, ter arhivi virtualnih sistemov, kot so VMware virtualni sistemi.
8. Kateri seji pripada, vsaka seja lahko zajema več objektov.
9. Koliko časa se podatki hranijo; v sekundah koliko časa se hranijo.
10. Ime gostitelja (hostname); od kod izvirajo podatki, ki se shranjujejo.
11. Stopnja kompresije; nekatere naprave podpirajo stiskanje podatkov, shranimo informacijo, koliko so bili podatki iz tega objekta stisnjeni od 1 (nič kompresije) do 0 (popolna kompresija).

Poleg informacij o sejah in objektih imamo še podatke o napravah, ki opravljajo arhiviranje. To so lahko različni NAS in SAN sistemi, oblačne rešitve in zapisovalci na magnetne trakove.

Informacije imamo tudi o medijih, na katere se shranjujejo podatki, torej o diskih ter kasetah. Za vsakega vemo, kolikšen je skupen prostor, kolikšna

je zasedenost, kdaj je bil medij prvič vstavljen v nas sistem, ter kdaj smo nanj nazadnje pisali in kdaj iz njega brali.

Imamo informacije, kdaj in na kakšen način se kaj arhivira, ker se večina arhiviranj požene glede na urnik.

3.2.3 Raziskava podatkov

Produkcijsko okolje je veliko, zato je tudi količina podatkov, ki smo jo pridobili velika, čeprav je časovno obdobje analize dokaj kratko.

Za domače okolje imamo veliko manjšo količino podatkov, samih instanc podatkov je manj za več kot dvestokrat. Ampak ti podatki obsegajo dvakrat daljše obdobje. Vse to je dobro razvidno iz tabele 3.1.

	domače okolje	produkcijsko okolje
časovni okvir	1.1.2016 - 1.7.2016	1.12.2015 - 10.3.2016
število sej	1613	538314
število objektov	4925	5662197

Tabela 3.1: Količina podatkov na katerih smo opravili analizo.

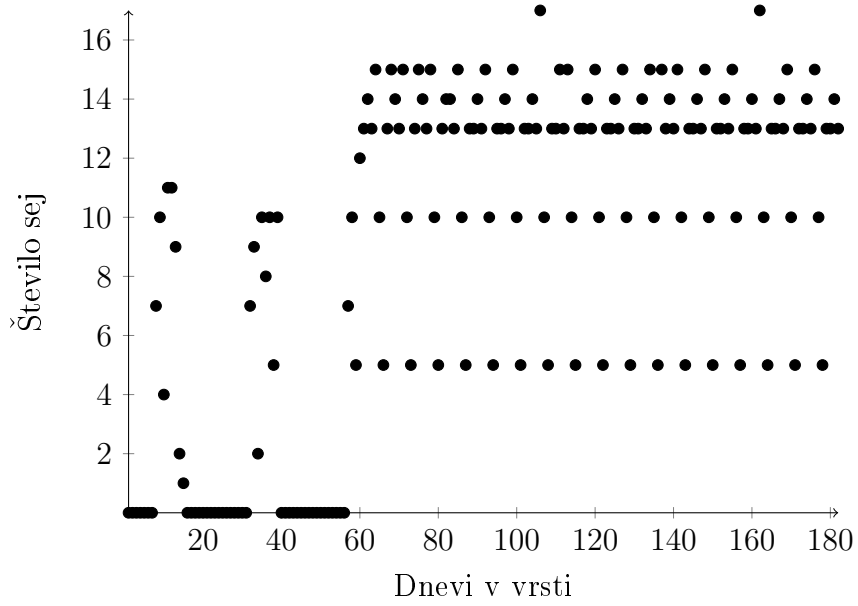
Za nas pomembni podatki so velikost objektov in vse informacije o času, začetek, konec ter koliko časa se podatki hranijo. Ostali podatki za trenutno analizo niso pomembni. S pomočjo njih bolje razumemo okolje in so nam na voljo, ko se odločimo razširiti analizo.

3.2.4 Preverba kvalitete podatkov

Pomembno pri podatkih je, da ni manjkajočih obdobij, saj za analizo potrebujemo čim bolj popolno časovno vrsto.

3.2.5 Priprava podatkov

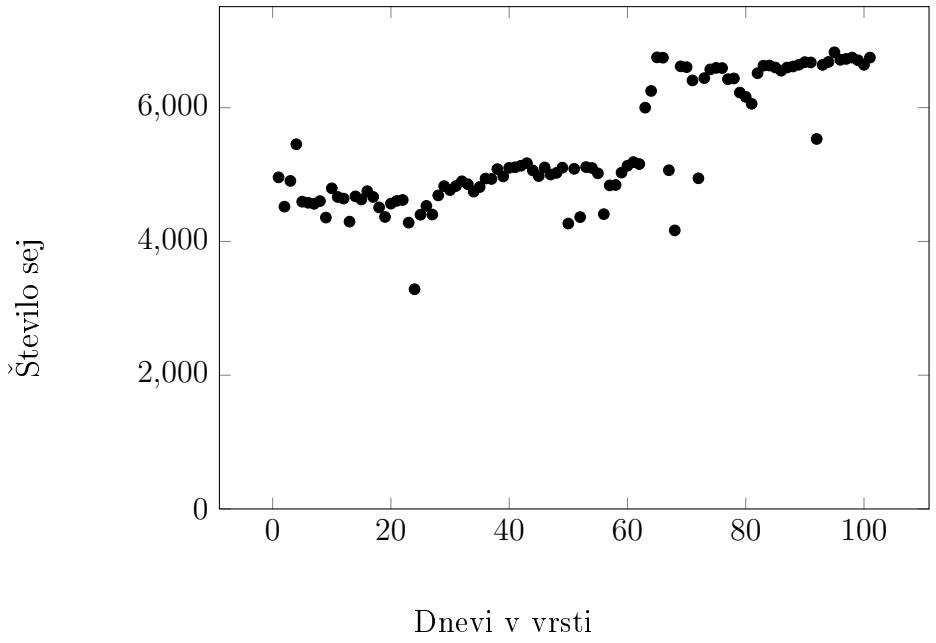
Grafa 3.1 in 3.2, prikazujeta število sej na dan.



Slika 3.1: Ševilo sej na dan, domače okolje

Kot lahko vidimo iz slike 3.1, imamo v domačem okolju na začetku nekaj lukenj, dni brez sej. Ne vemo, ali je to posledica vzdrževalnih del ali smo imeli težave pri zbiranju podatkov, zato smo se odločili, da teh podatkov ne vključimo v analizo, torej upoštevamo le podatke od 1. marca 2016 do 1. julija 2016.

Opazimo periodičnost sej. Arhivske seje se ponavadi izvajajo samodejno po določenemu periodičnemu urniku. Iz grafa se lepo vidi, da se arhiviranje podatkov v podjetju izvaja avtomatsko.



Slika 3.2: Število sej na dan, produkcijsko okolje

V podatkih iz produkcijskega okolja nimamo lukenj, zato smo uporabili celoten interval podatkov. Iz slike 3.2 lahko vidimo, da je število dnevnih sej dvignilo s 5000 na skoraj 7000.

Vidimo nekaj točk, ko je bilo število sej dosti manjše od okoliških dni. Te po vsej verjetnosti nakazujejo na dneve, ko so bile v okolju težave, ali pa se je opravljalo vzdrževanje sistemov.

3.2.6 Izgradnja ter integracija podatkov

Pripraviti moramo podatke za podatkovno rudarjenje. Povezati želimo seje in objekte. Želimo ustvariti časovno vrsto.

Za vsak dan smo sešteli vse objekte, ki so še pod arhivom. Da je objekt še pod arhivom, pomeni, da je čas, ko smo ga shranili, seštet s časom, koliko časa bomo hranili objekt, večji od trenutnega datuma.

Priprava podatkov v produkcijskem okolju po opisanem postopku je trajala deset minut, glede na trende rasti bo ob morebitni postavitvi okolje dva do trikrat večje. Ta čas je že toliko dolg, da ga je treba upoštevati ob vpeljavi

v produkcijo.

3.3 Modeliranje

Najprej opišemo uporabljene metode. Potem pripravimo načine, kako bom preizkusili točnost napovedi in zapišemo njihove matematične formule in jih opišemo. Modele tudi zgradimo in izračunamo ocene točnosti.

3.3.1 Izbira modelirne tehnike

3.3.1.1 Linearna regresija

Izbrali smo modeliranje po postopku linearne regresije. Model linearne regresije lahko opišemo z enačbo (3.1).

$$y = b_0 + b_1 * x \tag{3.1}$$

b_0 je začetna vrednost.

b_1 je naklon premice.

$y\ in\ x$ sta slučajni spremenljivki.

Razlika med resnično vrednostjo in modelom linearne regresije je napaka. Lahko jo predstavimo kot ϵ.

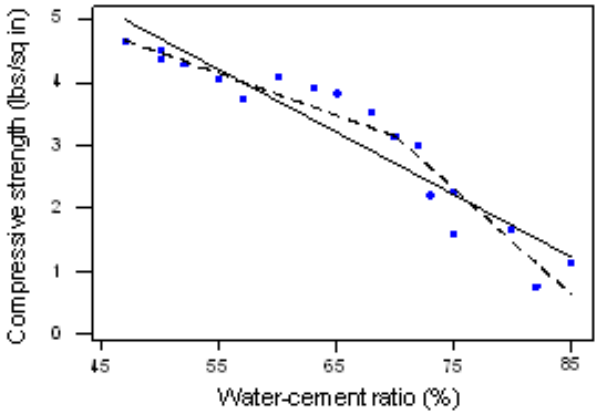
$$y = b_0 + b_1 * x + \epsilon \tag{3.2}$$

b_1 oziroma naklon lahko interpretiramo kot spremembo vrednosti y za spremembo vrednosti x. [4]

Da je linearna regresija uspešna, mora biti razmerje med x in y linearno. Linearna regresija ima težave, če so dogodki nepovezani, ali pa imamo ostro spremembo v dogodkih.

3.3.1.2 Linearna regresija po kosih

Linearna regresija po kosih je nadgradnja navadne linearne regresije. Namesto da linearni model izračunamo na vseh podatkih hkrati ga izračunamo večkrat na kosu podatkov.



Slika 3.3: Primer linearne regresije po kosih [13].

Iz slike 3.3 je razvidno da lahko linearna regresija po kosih predstavlja boljši približek podatkov kot bi ga navadna linearna funkcija.

3.3.1.3 K-najbližjih sosedov

Metoda k-najbližjih sosedov kot model hrani svoje učne primere. Za nov primer poišče algoritem k- najbližjih, podobnih primerov, in oceni verjetnostno porazdelitev iz relativne porazdelitve teh najbližjih primerov. Deluje po principu, da imajo bližnji primeri podobne atribute, kot točka napovedi. Če želimo predvideti obnašanje novega primera, pogledamo njegove najbližje sosede [5].

3.3.2 Izdelava načrta za testiranje

3.3.2.1 Način izračuna

Modele zgrajene po pristopu linearne regresije smo testirali na učni množici. Vzeli smo vse podatke, ki smo jih uporabili za izgradnjo modela in na njih preizkusili napovedi.

Modele zgrajene po pristopu k-najbližjih sosedov bomo testirali na delu podatkov. Podatke bomo razdelili na učno in testno množico. Učna množica bo obsegala 66 % podatkov in testna množica preostale podatke.

3.3.2.2 Mere napake

Povprečna absolutna napaka [7]:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \tag{3.3}$$

Koren srednje kvadratne napake [7]:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}} \tag{3.4}$$

Za oba napaki, MAE in RMSE bomo izračunali relativno vrednost tako da jih bomo delili s povprečno vrednostjo.

Ter koeficient R^2

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}},\ SS_{res} = \sum_{i=1}^{n}(\hat{y}_i - y_i)^2,\ SS_{tot} = \sum_{i=1}^{n}(y_i - \bar{y_i})^2 \tag{3.5}$$

y_i predstavlja pravilno vrednost.

$\hat{y}_i$ predstavlja vrednost, ki smo jo predvideli.

$\bar{y_i}$ predstavlja povprečno vrednost vseh instanc.

n predstavlja število teh parov.

Koeficient R^2 lahko zavzema vrednosti med 0 in 1, kjer vrednost bližje

1 pomeni, da je povezanost med vhodnimi in izhodnimi spremenljivkami močnejša. Vrednost 0 pomeni da je najboljša napoved vodoravna premica oziroma povprečna vrednost parametrov [8].

4. Vrednotenje modelov

V tem poglavju vse modele izgradimo in za njih izračunamo mere napake. V nadaljevanju jih ovrednotimo z neposredno primerjavo napovedanih in pravilnih vrednosti na testnih primerih. Vse uporabljene metode primerjamo med sabo in izberemo najbolj uspešno. Na koncu podamo ideje, kako bi naše delo nadgradili.

4.1 Izgradnja modelov in izračun uspešnosti

S pomočjo orodja WEKA smo izgradili modele linearne regresije. Formule modelov so vidne v tabeli 4.1.

	domače okolje	produkcijsko okolje
vsi učni primeri	y=131.3*x+13204.7	y=6074.5*x+4424271.5
zadnjih 20 učnih primerov	y=-118.2*x+22430.7	y=0*x+5150230.4

Tabela 4.1: Modeli linearne regresije.

V tabeli 4.2 prikazujemo izračune napak za modela linearne regresije. Za domače okolje dobimo slabe rezultate, največja napaka je kar 240 %.

	domače okolje	produkcijsko okolje
relativni MAE	23,4 %	5,8 %
relativni RMSE	27,6 %	6,8 %
največja napaka	240 %	33 %
R^2 testnih primerov	0,38	0,25

Tabela 4.2: Napake linearne regresije.

V tabeli 4.3 prikazujemo izračune napak za modela linearne regresije po kosih. Za obe okolij dobimo dobre rezultate, največja napaka v produkciji je le 4 %, v domačem okolju pa je 11 %.

	domače okolje	produkcijsko okolje
relativni MAE	3,9 %	1,3 %
relativni RMSE	5,1 %	1,6 %
največja napaka	11 %	4 %
R^2 testnih primerov	0,28	0,04

Tabela 4.3: Napake linearne regresije po kosih.

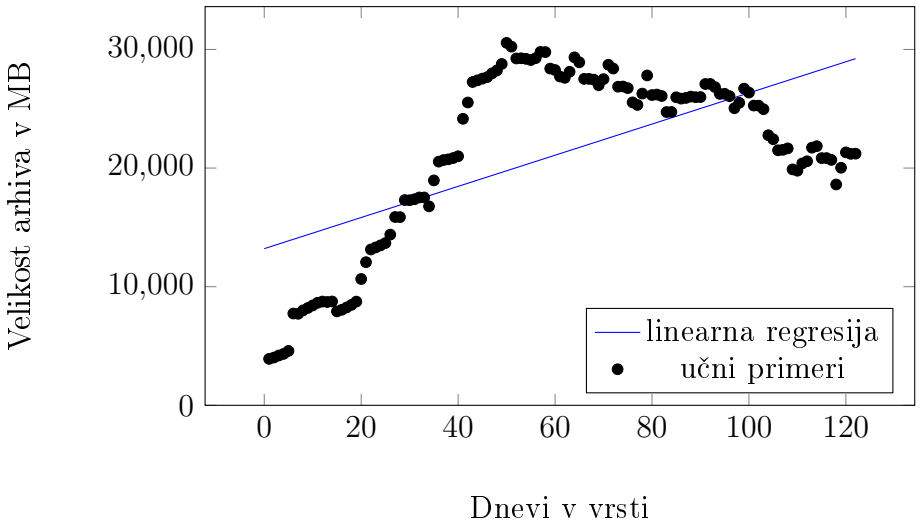
V tabeli 4.4 prikazujemo izračune napak za modela k-najbližjih sosedov. Za domače okolje rezultati niso zadovoljivi z 39 % največjo napako. V produkcijskem okolju so napovedi najboljše od uporabljenih metod z le 3 % največjo napako.

	domače okolje	produkcijsko okolje
relativni MAE	18,4 %	0,7 %
relativni RMSE	20,5 %	1,1 %
največja napaka	39 %	3 %
R^2 testnih primerov	0,73	0.06

Tabela 4.4: Napake k-najbližjih sosedov.

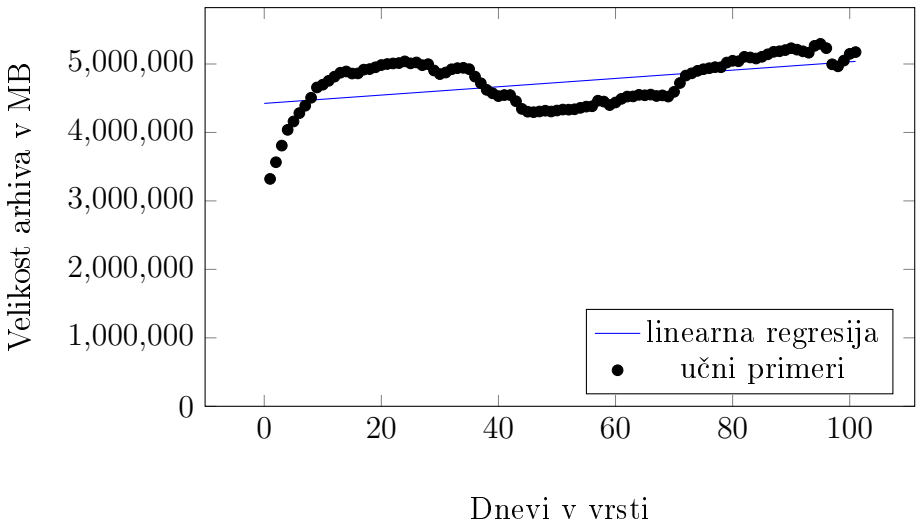
4.2 Vrednotenje rezultatov modela linearne regresije

Model linearne regresije ne zadovolji naših poslovnih ciljev, saj so v povprečju napovedi preveč netočne, da bi lahko iz njih ugotovili, kdaj bo zmanjkalo prostora ali potreb po pospešitvi podatkovnih povezav.



Slika 4.1: Linearna regresija čez vse podatke v domačem okolju.

Linearni model je popolnoma netočen, kar je dobro razvidno iz grafa 4.1. Ne zazna sprememb v trendu, kot je sprememba, iz rasti v padanje velikosti po 50. primeru.



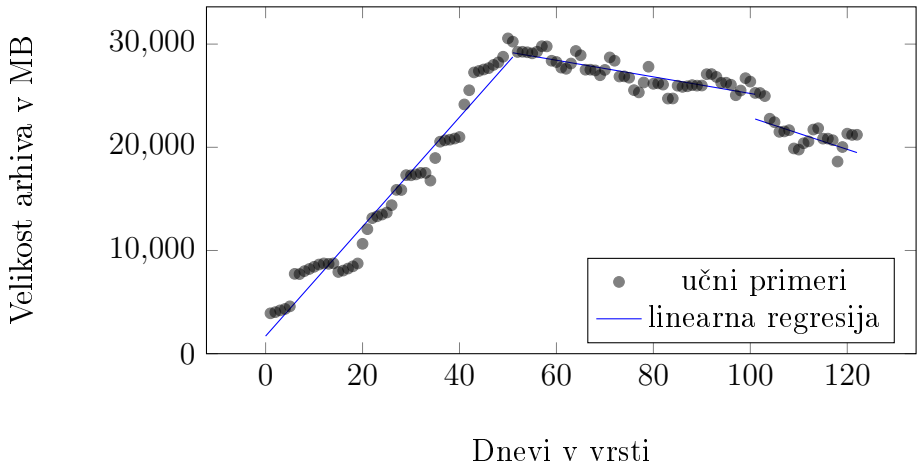
Slika 4.2: Linearna regresija čez vse podatke v produkcijskem okolju.

Iz grafa 4.1 vidimo, da model za zadnje instance prikaže dobre napovedi, a je daljša obdobja netočen.

Ker se trendi v obeh okoljih spreminjajo, linearna regresija čez daljša obdobja ne nudi dobrih napovedi.

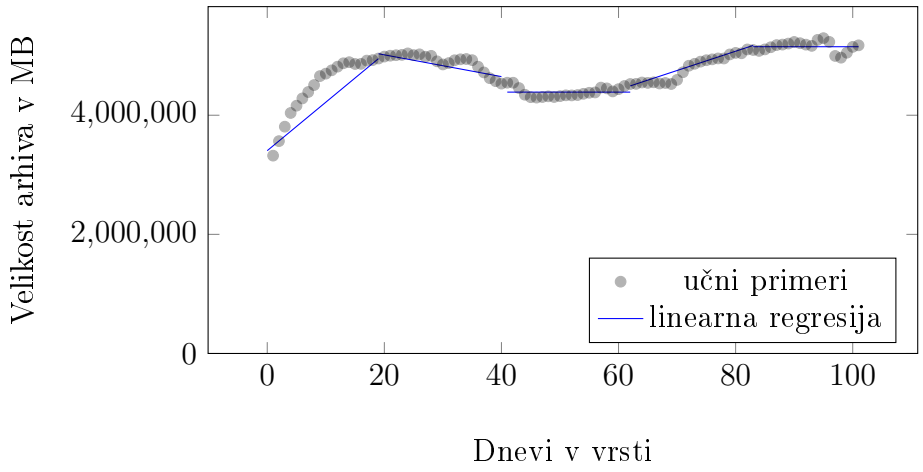
4.3 Vrednotenje modela linearne regresije po kosih

Zgradili smo modele po kosih in izbrali tistega, ki dosega najboljše rezultate. Model mora biti zgrajen na vsaj 20 učnih primerih, s tem preprečimo da bi se preveč prilagajali podatkom oziroma da bi lokalni dogodek vplival na napoved globalnih trendov.



Slika 4.3: Linearna regresija po kosih v domačem okolju.

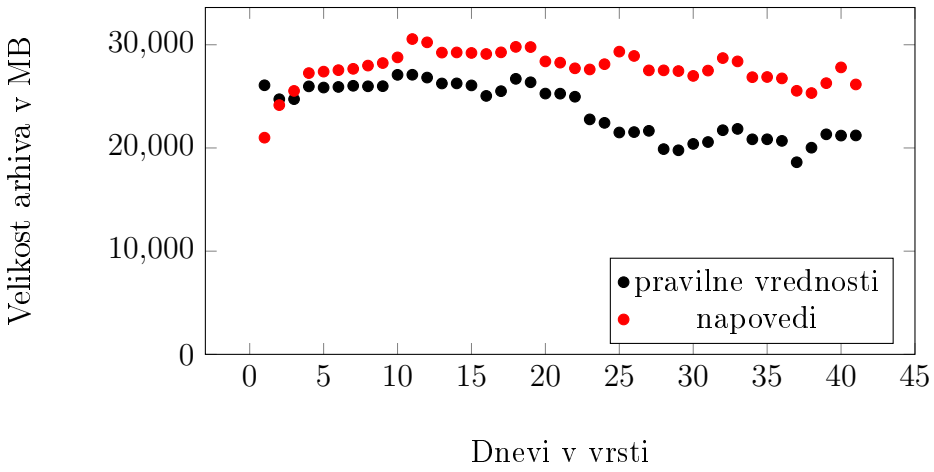
Model iz grafa 4.3 je padajoča linearna funkcija, ki daje solidne napovedi. Podatki kažejo tri linearne trende, trend hitre rasti od začetka do 50. primera, potem imamo do 102. primera trend počasnega padanja, podatki se končajo s podobnim trendom padanja, le da imamo na začetku instanten večji padec. Ta dogodek pomeni, da je IT administrator zbrisal podatke iz arhiva ročno.



Slika 4.4: Linearna regresija po kosih v produkcijskem okolju.

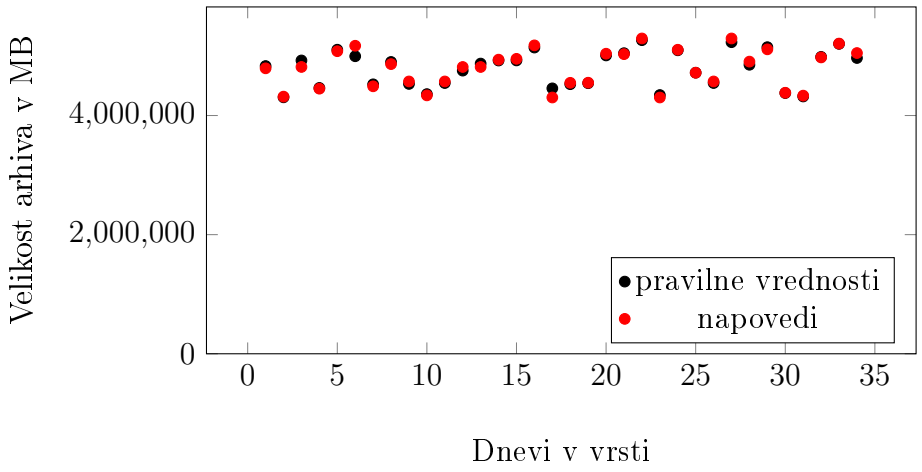
Model iz grafa 4.4 predstavlja vodoravna ravna črta, ki daje dobre napovedi. Podatki kažejo pet linearnih trendov, rast, padec, ustalitev, ponovno rast in ponovno ustalitev. Graf nakazuje, da se trendi ponavljajo v časovnih presledkih. Za potrditev te teze, žal nimamo podatkov iz dovolj dolgega časovnega obdobja.

4.4 Vrednotenje modela k-najbližjih sosedov



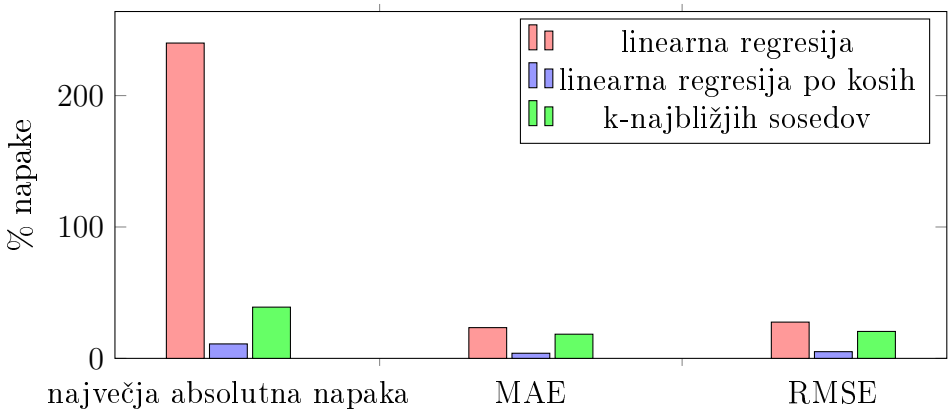
Slika 4.5: K-najbližjih sosedov v domačem okolju.

Napovedi trendov velikosti so dokaj slabe. V zadnjem delu se vidi skok z grafa 4.1, imamo večji padec kapacitete, ki se zgodi v enem dnevu, ki ga metoda najbližjih sosedov ni predvidela. Najbližje sosede smo iskali na podlagi časovne razdalje. Prvih nekaj napovedi je dobrih, ko pa se časovna razdalja veča se z njo povečuje tudi napaka, kar je razvidno iz grafa 4.5. Iz tega vidimo, da je metoda k-najbližjih sosedov dobra za napovedi za naslednjih nekaj dni, ne pa dlje v prihodnost.

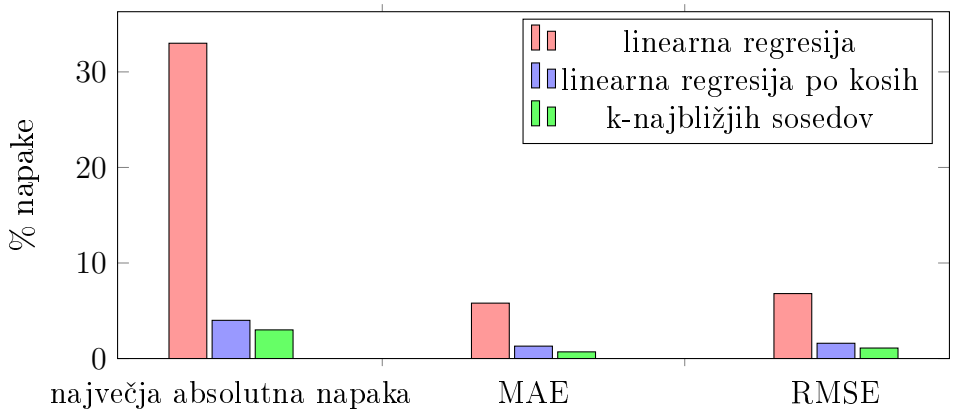


Slika 4.6: K-najbližjih sosedov v produkcijskem okolju.

Napoved velikosti podatkov je zelo dobra, iz grafa 4.6 vidimo, da v tem okolju ne prihaja do velikih sprememb, kot v okolju, ki ga imamo v podjetju.



Slika 4.7: Mere napake v domačem okolju.



Slika 4.8: Mere napake v produkcijskem okolju.

Grafa 4.7 in 4.8, ki prikazujeta relativno MAE, relativni RMSE in največjo absolutno napako za obe okolji, prikažeta, da je navadna regresija najslabša izmed preizkušenih modelov. Linearna regresija po kosih daje boljše rezultate. Iz rezultatov pa tudi vidimo, da se modeli v različnih okoljih ne obnašajo enako, algoritem k-bližnjih sosedov daje v produkcijskem okolju izmed treh algoritmov najboljše rezultate, medtem ko je v našem domačem okolju dokaj šibek.

Modeli niso popolni, ustaljeno dobre rezultate daje linearna regresija po kosih, ki se najbolj približa zadanemu cilju maksimalno 10 % absolutne napake. V domačem okolju je maksimalna napaka 11 % in v produkcijskem okolju 4 %. Metoda k-najbližjih sosedov je preveč odvisna od okolja, čeprav tega ne moremo zagotovo potrditi s samo dvema okoljema. Je pa v produkcijskem okolju dosegel najboljši rezultat s samo 3 % absolutno napako. V tem trenutku je izbira najboljšega modela težka, zato bomo modele sprotno ocenjevali v produkciji.

4.5 Pregled opravljenega dela

Pridobili smo podatke iz enega naših največjih produkcijskih okolij, ampak da bi lahko z večjo zanesljivostjo rezultate vpeljali v produkcijo, bi morali le-te preizkusiti na več različnih okoljih. Upravljavce okolji za arhiviranje

podatkov bi morali bolj vključili v analizo podatkov. Ob najdenih anomalijah ali hitri spremembi trendov bi jih morali povprašati o ozadju, kar bi nam omogočilo boljše razumevanje podatkov in prilagoditev algoritmov.

Več časa bi morali nameniti preizkušanju algoritmov, preizkusiti bi morali več različnih algoritmov. Ta del analize je bil časovno pod dimenzioniran.

Za naše potrebe orodje WEKA zadostuje. V času uporabe smo ugotovili, da orodje omogoča širino, saj implementira več različnih klasifikatorjev za podatkovno rudarjenje. Trenutno pa ga v produkciji ne smemo uporabiti, ker je licenciran pod GNU General Public license (GPL). Licenca zahteva, da so produkti, ki uporabljajo dele, ki so licencirani pod GPL, tudi sami licencirani pod GPL. [11] To pomeni, da naš program, ki je lastniški, ne sme uporabljati WEKA knjižnice. Če bi ga še vedno želeli uporabiti, WEKA omogoča pridobitev komercialne licence. [18]

4.5.1 Potrebne izboljšave

Obstoječo analizo je potrebno izvesti na podatkih iz več različnih okolji, saj se modeli ne obnašajo enako. Uporabili bomo več različnih algoritmov in videli, kako se obnašajo. Tako bi lahko izbirali najboljšo metodo, glede na začetno testiranje v danem okolju.

Oskrbnike arhivskih okolji moramo bolj vključili v analizo podatkov. Ob najdenih anomalijah ali hitri spremembi trendov bi jih morali povprašati o ozadju, kar bi nam omogočilo boljše razumevanje podatkov in prilagoditev algoritmov.

4.6 Uvajanje v produkcijo

4.6.1 Plan za uvedbo v produkcijo

Uvedbo v produkcijo lahko razdelimo na tri korake:

1. Pripravo podatkov, podatke imamo že zbrane, moramo pa opraviti postopke priprave in agregacij. Na podoben način, kot pri pripravi po-

datkov za podatkovno rudarjenje, jih lahko pripravimo v produkciji.

Razlika je, da jih bomo v produkciji računali vsak dan na novih podatkih. Ob koncu dneva bomo obstoječim podatkom dodali nove, s pomočjo sprožilcev [15].

2. Nato izgradimo modele.

3. S pripravljenimi modeli pripravimo poročilo z napovedjo rasti in napovemo, kdaj bo zmanjkalo prostora v okolju za arhiviranje. Opozorimo uporabnika na točnost napovedi.

4.6.2 Plan za spremljanje in vzdrževanje

V ozadju bomo zbirali podatke o napovedih skozi čas in njihovi natančnosti, ki jih bomo zbirali od čim več uporabnikov. V program bomo integrirali logiko za zbiranje teh podatkov.

5. Zaključek

V diplomskem delu smo po postopku podatkovnega rudarjenja CRISP-DM analizirali probleme napovedovanja trendov pri rezervnem shranjevanju podatkov. Za napovedovanje smo uporabili linearno regresijo, linearno regresijo po kosih in k-najbližjih sosedov.

Rezultati so pokazali, da model linearne regresije ne daje dobrih rezultatov. Preostala modela dajeta solidne rezultate, a je linearna regresija po kosih bolj konsistentna. Napovedi so dovolj dobre, da lahko pomagajo upravljavcu okolja pri odločitvah, niso pa dovolj zanesljive, da bi se lahko slepo zanesli nanje.

Za najbolj pomembna koraka podatkovnega rudarjenja sta se izkazala zbiranje in priprava podatkov. Okolja za arhiviranje podatkov se lahko med sabo zelo razlikujejo, čeprav za osnovo uporabljajo isti program, saj so odvisna od potreb uporabnika.

Delo bi bilo smiselno nadgraditi z boljšimi podatki, predvsem njihovo količino in raznovrstnostjo. V program za nadziranje okolja za arhiviranje bi bilo pametno dodati logiko za samodejno pripravo podatkov, ki bi jih lahko uporabili za pripravo modelov. Smiselno bi bilo razširiti število preizkušenih modelov. V analizo bi bilo smotrno dodati vrsto okolja in glede na to prilagoditi podatkovno rudarjenje. Za to bi morali najprej pripraviti množico značilnih okolij in določiti njihove značilnosti.

Literatura

[1] A. Azavedo and M. F. Santos, "KDD, SEMMA and CRISP-DM: A Parallel Overvie", v zborniku *IADIS European Conference on Data Mining 2008*, Amsterdam, The Netherlands, jul. 2008, str. 182-185.

[2] M. Chamness, "Capacity Forecasting in a Backup Storage Environmen", v zborniku *LISA '11: 25th Large Installation System Administration Conference*, Boston, Massachusetts, dec. 2011, str. 12-12.

[3] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, R. Wirth, *CRISP-DM 1.0 Step-by-step data mining guide*, SPSS Inc., 2000.

[4] D. C. Montgomery, E. A. Peck, G. G. Vining, *Introduction to Linear Regression Analysis, Fifth Edition*, Wiley, 2012.

[5] S. Sumathi, S. N. Sivanandam, *Introduction to Data Mining and its Applications*, Springer, 2006.

[6] I. H. Witten, E. Frank, M. A. Hall, *DATA MINING Practical Machine Learning Tools and Techniques, Third Edition*, Elsevier, 2011.

[7] A. Bifet. Regression. http://www.cs.waikato.ac.nz/ abifet/523/Regression-Slides.pdf. [Elektronski] [Dostopano 19. 8. 2016].

[8] J. Frost. Regression Analysis: How Do I Interpret R-squared and Assess the Goodness-of-Fit?. http://blog.minitab.com/blog/adventures-in-

statistics/regression-analysis-how-do-i-interpret-r-squared-and-assess-the-goodness-of-fit. [Elektronski] [Dostopano 19. 8. 2016].

[9] Google Scholar. Top publications - Data Mining & Analysis. https://scholar.google.si/citations?view_op=top_venues&hl=en&vq=eng_dataminінganalysis. [Elektronski] [Dostopano 19. 8. 2016].

[10] Smart Vision Europe Ltd. Phases of the CRISP-DM reference model. http://crisp-dm.eu/. [Elektronski] [Dostopano 19. 8. 2016].

[11] Inc. Free Software Foundation. GNU General Public License. http://www.gnu.org/licenses/#GPL. [Elektronski] [Dostopano 19. 8. 2016].

[12] ManageEngine. Storage Capacity Forecasting and Planning. https://www.manageengine.com/products/opstor/storage-capacity-forecasting-planning.html. [Elektronski] [Dostopano 19. 8. 2016].

[13] I. Pardoe, L. Simon, D. Young. Regression Methods. https://onlinecourses.science.psu.edu/stat501/node/310. [Elektronski] [Dostopano 19. 8. 2016].

[14] G. Piatetsky. CRISP-DM, still the top methodology for analytics, data mining, or data science projects. http://www.kdnuggets.com/2014/10/crisp-dm-top-methodology-analytics-data-mining-data-science-projects.html. [Elektronski] [Dostopano 19. 8. 2016].

[15] The PostgreSQL Global Development Group. About. https://www.postgresql.org/about/. [Elektronski] [Dostopano 19. 8. 2016].

[16] Signature Technology Group. Capacity Management and Forecasting Best Practices and Recommendation. http://www.signaturetechnology.com/blog/capacity-management-and-forecasting-best-practices. [Elektronski] [Dostopano 19. 8. 2016].

[17] SolarWinds. Storage Capacity Planning. http://www.solarwinds.com/topics/storage-capacity-planning. [Elektronski] [Dostopano 19. 8. 2016].

[18] The University of Waikato. WEKA Wiki. https://weka.wikispaces.com/. [Elektronski] [Dostopano 19. 8. 2016].

[19] Wolfram Research, Inc.. Correlation Coefficient. http://mathworld.wolfram.com/CorrelationCoefficient.html. [Elektronski] [Dostopano 19. 8. 2016].